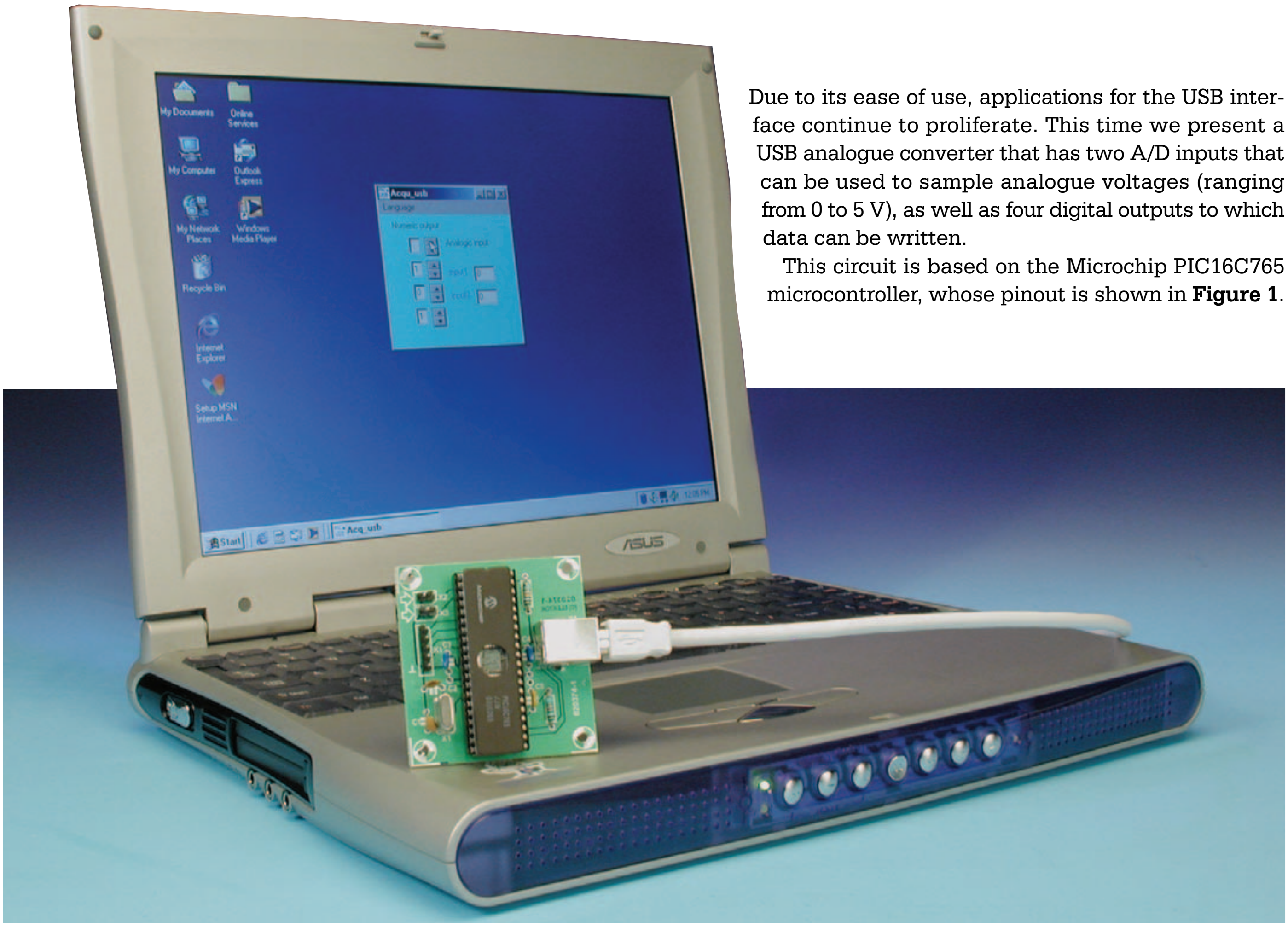
USB Analogue Converter

With two analogue inputs and four digital outputs

Design by E. Brador

This circuit is based on the PIC16C765, a recent addition to the large Microchip PIC family. It can be used to measure two analogue voltages (0–5 V) via two analogue inputs and send the results to a PC via a USB connection. There are also four digital outputs whose levels can be set using the PC.



Due to its ease of use, applications for the USB interface continue to proliferate. This time we present a USB analogue converter that has two A/D inputs that can be used to sample analogue voltages (ranging from 0 to 5 V), as well as four digital outputs to which data can be written.

This circuit is based on the Microchip PIC16C765 microcontroller, whose pinout is shown in **Figure 1**.

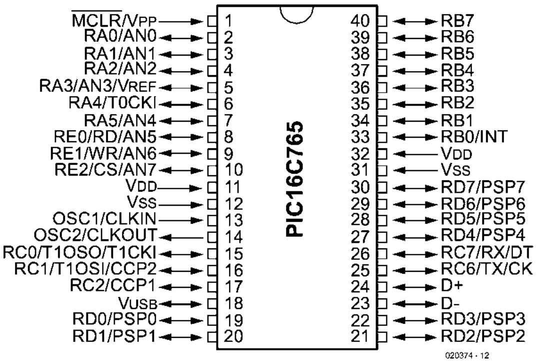


Figure 1. PIC16C765 pinout (source: Microchip).

The PIC16C765

Together with its little brother, the PIC16C745 (28 pins), the PIC16C765 was the first PIC processor to have an on-board USB interface. This RISC processor (with only 35 instructions) has 8 Kwords of 14-bit program memory and 256 bytes of data memory. It also features eight analogue inputs, a variety of digital inputs and outputs, a UART and a USB controller, among other things.

Microchip provides assembler files for this microcontroller in order to simplify the essential communications between the USB PIC and the PC. For more information, you can visit the manufacturer's website at http://www.microchip.com/

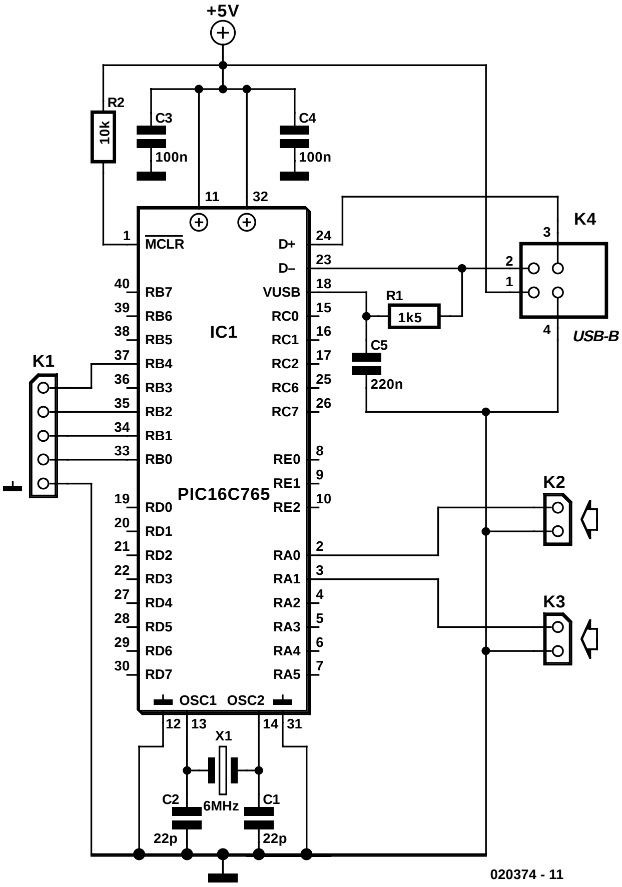


Figure 2. Except for IC1, there's not a lot of electronics in the USB analog converter.

Any electronic technician or hobbyist who wants to keep abreast of new developments can hardly avoid the USB port. Here we use USB Type 1 (USB 1.1), which is more than fast enough for sampling signals whose levels change relatively slowly, such as temperature, pressure, etc.

The electronics

A glance at the schematic diagram in **Figure 2** clearly shows that our circuit is almost unbelievably simple. The microcontroller is surrounded by only a handful of components, consisting primarily of the quartz crystal and the usual capacitors connected to inputs OSC1 and OSC2, which collectively ensure that the program runs at 24 MHz inside the IC, despite the fact that a 6-MHz crystal is used. This is because an integrated PLL causes the basic frequency generated by the crystal to be multiplied by a factor of four.

Pushbutton switch S1 can be used to reset the processor in the event of problem. The presence of a device on the USB port is detected via resistor R1 (1k5). The rest of the circuit consists of various connectors for the inputs and outputs.

One of the advantages of USB can already be seen here: the circuit draws its power directly from the USB port.

The circuit board

Although a printed circuit board has been designed for this circuit (**Figure 3**), it is naturally not much larger than the IC. Considering the small number of components (a microcontroller, a USB chassis-mount connector, a crystal, two resistors and five capacitors), instructions for assembling the circuit board are actually unnecessary, except that we recommend using a good-quality socket for the microcontroller.

If you cannot program the PIC16C765 used in this circuit, a pre-programmed version can be ordered form Readers Services under order number **020374-41**.

Installation

After the USB cable (which also provides the power) has been connected, the circuit will be recognised as a Human Interface Device (HID). However, this only works properly under Windows 98 SE, Windows ME, Windows XP or Windows 2000. After the device has been recognised, the proper driver will be installed.

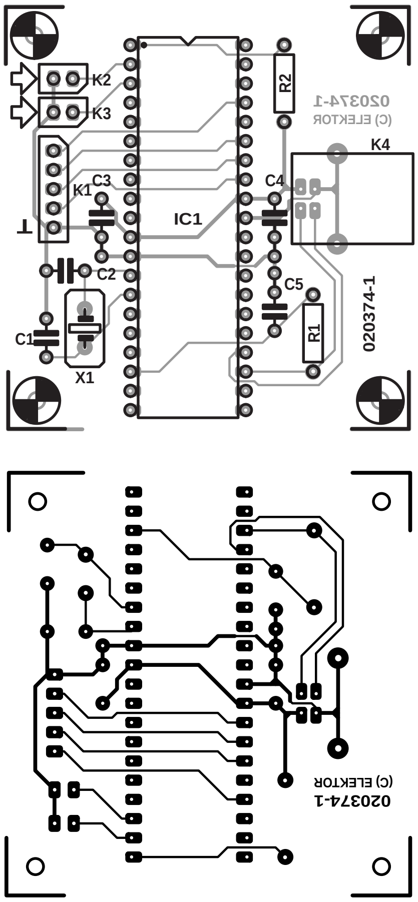
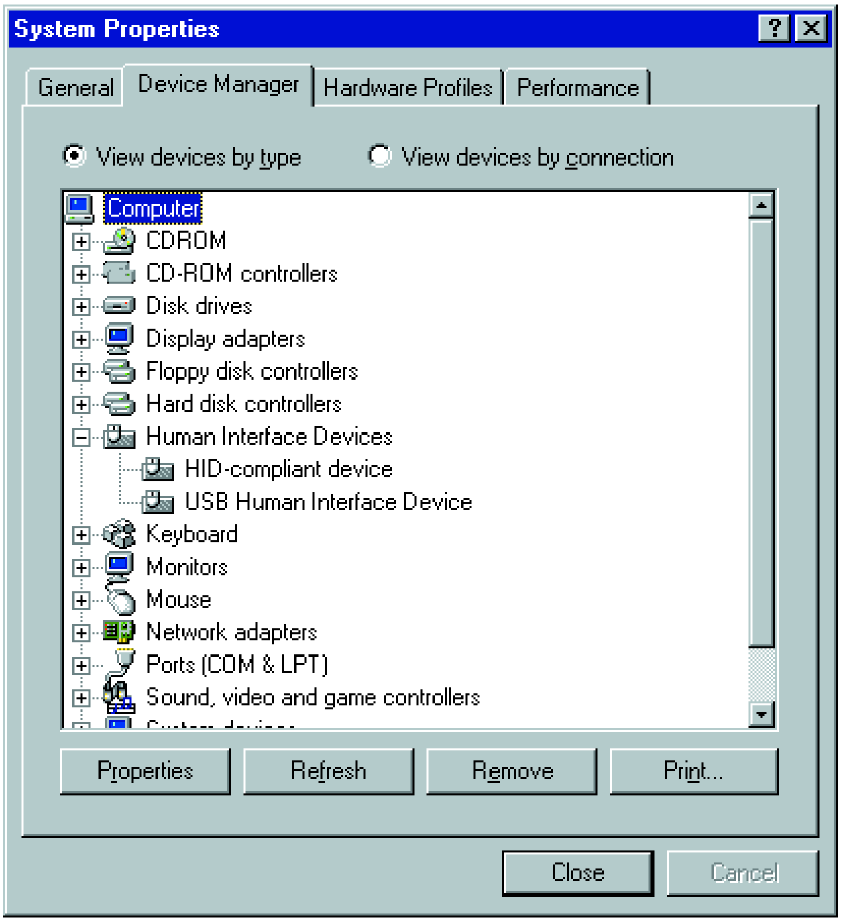


Figure 3. Printed circuit board and component layouts for the USB converter circuit.


System Properties

General | Device Manager | Hardware Profiles | Performance

● View devices by type ○ View devices by connection

Computer
- CDROM
- CD-ROM controllers
- Disk drives
- Display adapters
- Floppy disk controllers
- Hard disk controllers
- Human Interface Devices
  - HID-compliant device
  - USB Human Interface Device
- Keyboard
- Monitors
- Mouse
- Network adapters
- Ports (COM & LPT)
- Sound, video and game controllers

Figure 4. Windows recognises the interface as an HID (Human Interface Device). This screendump is taken from Windows XP.

COMPONENTS LIST

Resistors:
R1 = 1kΩ5
R2 = 10kΩ

Capacitors:
C1,C2 = 22pF
C3,C4 = 100nF
C5 = 220nF

Semiconductors:
IC1 = PIC16C765, programmed, order code **020374-41**

Miscellaneous:
X1 = 6MHz quartz crystal
K1 = 4-way PCB terminal block
K2,K3 = 2-way PCB terminal block
K4 = USB socket, PCB mount, type B
PCB, order code **020374-1**
Disk, contains hex-code and Windows-software, order code **020374-11** or Free Download

The microcontroller identifies itself to the PC by specifying its vendor ID (VID) and product ID (PID). As this is a hobbyist circuit, we have retained the default settings from Microchip. If you consider that up to 256 different devices can be connected to a single USB port, you can understand the need for unambiguous identification.

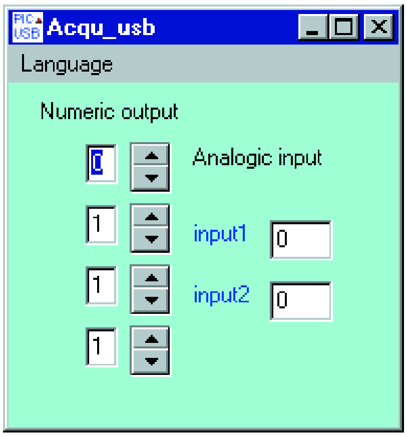
After being installed, the circuit should be visible as a USB device in the Device Manager window (see the screen dump in **Figure 4**).

The assembler program

At the Microchip website, you can find a variety of programs that illustrate how to program this microcontroller for USB communications, which means for data traffic via the USB port.

If you want to write programs for this microcontroller, we advise you to first become quite familiar with the USB protocol.

Some of the important portions of the program are printed in **Listings 1 & 2**, so you can see what is involved


Acqu_usb

Language

Numeric output

Analogic input: 0
1 input1 0
1 input2 0
1

Figure 5. The program window shows the values measured on the two analogue inputs and the levels of the four digital outputs.

in making an analogue measurement and sending the measured values.

Assembling the program is more complex than with a PIC16F84, since several files from Microchip are necessary for this purpose. The program can be compiled using MPLAB, the well-known PIC development environment that Microchip makes available free of charge.

The microcontroller program is available in the form of a hex file (Elk.hex) and is available on diskette from Readers Services (order number **020374-11**) or from the *Elektor Electronics* website.

Operation

The PC software for the PIC is written in Delphi, and some of its subroutines are suitable for use with USB.

It is necessary to install the file port.dll in the same directory as the main program.

The compiled PC program, which is called Acq_usb, has been tested under Windows XP, Windows ME and Windows 98 SE, and it works perfectly under these three operating systems.

Using this program, voltages between 0 and 5 V can be cyclically sampled on two channels (see the screen dump in **Figure 5**). The program automatically starts sampling.

To test the circuit, you can use a power supply or battery with a potentiometer to allow the voltage to be adjusted over the range of 0–5 V.

The arrow buttons can be used to write ‘1’ of ‘0’ values to the four outputs by clicking on the up arrow (for ‘1’) or the down arrow (for ‘0’).

Although this is a ‘bare bones’ circuit, it can be used for a wide variety of measurements by connecting various types of converters or transducers to the inputs (via an interface if necessary). In this way, you can use it to measure temperatures or pressures.

(020374-1)

Listing 1. Analogue measurement subroutine (one channel).

```
acqudat
    banksel PIE1           ;memory bank selection
    bcf PIE1,ADIE          ;don't use interrupt for conversion

    banksel ADCON0
    bcf ADCON0,CHS2        ;input selection
    bcf ADCON0,CHS1        ;microprocessor's
    bcf ADCON0,CHS0        ;RA0


    bsf ADCON0,ADON        ;set internal converter


    pagesel Pdelay
    call Pdelay            ;call delay for conversion

    banksel ADCON0
    bsf     ADCON0,GO      ;start new conversion

loop
    btfsc   ADCON0,GO_DONE ;loop

    goto loop              ;conversion not yet finished

    banksel PIR1
    BCF PIR1,ADIF          ;reset end of conversion flag

    banksel ADRES          ;fetch conversion result
    movf ADRES,w
    banksel buffer
    movwf buffer           ;and move to buffer for transmission to PC
```

Listing 2. Subroutine for transmitting measured values to the PC.

```
SendData              ;subroutine for transmission of values to PC
  pagesel  PutEP1
  bankisel buffer     ;place buffer address into FSR
  movlw    buffer
  movwf    FSR
  movlw    0x4      ;send 4 bytes
  call     PutEP1
  btfss   STATUS,C ;stay in this loop until data transmission complete
  goto sendData

  banksel buffersFilled
  clrf     buffersFilled
  bankisel buffer
  clrf     buffer
  clrf     buffer+1
  clrf     buffer+2
  clrf     buffer+3
  clrf     buffer+4

  pagesel acqudat
  goto acqudat
```